White Paper

Integrating LSF Storage-Aware Plug-in with Operations Manager

Saif Hameed, NetApp
December2011| WP-7150

EXECUTIVE SUMMARY

This white paper provides an overview of how the LSF Storage-Aware Plug-in is integrated into the NetApp® environment. The plug-in provides LSF with the ability to schedule jobs based on the thresholds and criteria defined by customers. This white paper not only outlines the specifications, capabilities, and limitations of the plug-in but also the business value for customers.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 HISTORY

Platform LSF is one of the most powerful workload managers for demanding, distributed, and mission-critical high-performance computing environments. It provides a complete set of workload management capabilities, all designed to work together to address your high- performance computing needs. Platform LSF includes a comprehensive set of intelligent, policy-driven scheduling features that enable you to fully utilize your compute infrastructure resources.

In complex server farm environments typical of large and medium-sized semiconductor design environments, Platform LSF plays a key role in optimizing the use of computing resources and access to shared software licenses. Access to fast shared NFS storage, however, is often a critical bottleneck in these environments. With designs becoming more complex every day, and with limited shared storage capacity and bandwidth available, storage-related job failures can directly impact project schedules and time to market for new products.

NetApp, working along with Platform Computing, has developed a custom solution designed to integrate the Data ONTAP® operating environment with Platform LSF. This solution is aimed at preventing job failures due to storage bottlenecks. The integrated solution exposes Data ONTAP accessible load metrics on the NetApp system as LSF resources so that scheduling and policy decisions can be automated based on NFS system load. The solution uses LSF plug-in scheduler technology to minimize the impact on the LSF production environment.

2 SOLUTION OVERVIEW

NetApp Data ONTAP integration with the Platform LSF environment is designed to help you get the most out of your server farm by automatically throttling job submissions when critical thresholds associated with appliance performance are being exceeded. By using this approach it is possible to avoid cases in which jobs fail due to NFS time-outs and other storage subsystem failures. The integrated solution has the following features:

- The integrated solution monitors file system capacities and system bandwidth by interacting directly with Data ONTAP.
- Job dispatch rates are optimized based on storage capacity, network, and file system I/O metrics and bandwidth.
- The solution will map mount points to support multiple physical storage systems.

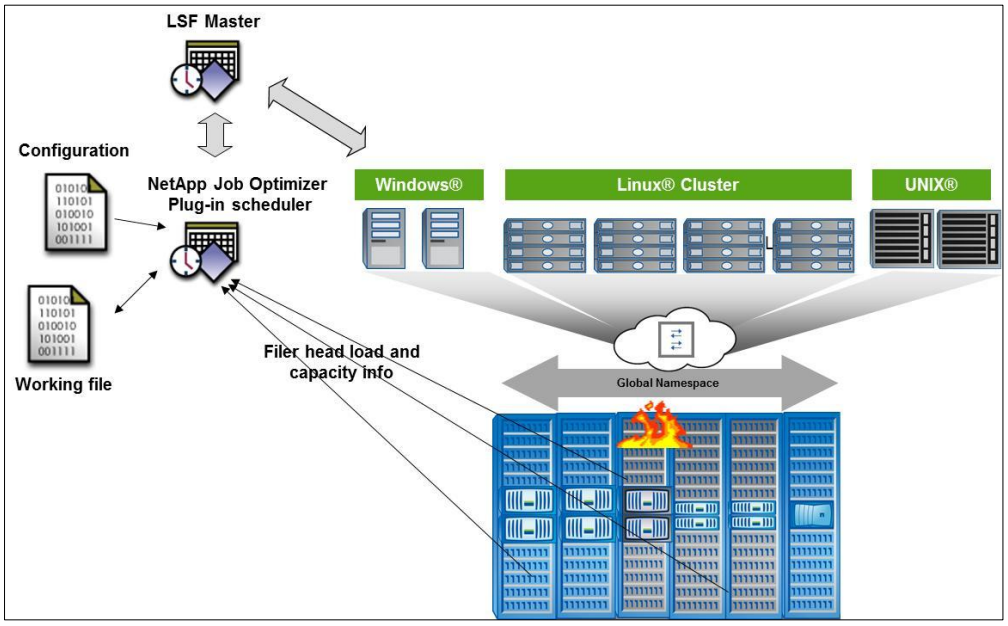
KEY BENEFITS

- Avoid oversubscription of storage controller.
- Suspend/hold jobs to prevent hot-spot intensity.
- Prevent jobs from encountering storage-related failures, enabling project deadlines to be met.
- Optimize job throughput according to available storage bandwidth and capacity.
- Incorporate the ability to capture storage-related performance metrics correlated with standard LSF reporting.
- Incorporate storage-related metrics as another resource for LSF to optimize as it does with CPUs and software licenses.

SOLUTION DESIGN

The LSF Storage-Aware Plug-in can be conceptualized as residing between the LSF and the Data ONTAP operating environment. Essentially, the plug-in gets information about the storage resources from the storage system via Operations Manager and makes it available to the LSF elim module. At this point, each job verifies the available resource thresholds before LSF executes the job. The thresholds are defined in the configuration file for the plug-in as described later in this paper.

Figure 1) Graphical representation of the solution.



3 MINIMUM REQUIREMENTS

Prerequisites include the following:

- Platform Computing LSF version 7u6 or later; 8.0 preferred
- NetApp Data ONTAP version 7.3.3 or 8.0+ (7-Mode only)
- NetApp Operations Manager version 4.0 or later
- LSF must be installed and running properly
- A compiler must be installed on one of the LSF machines that matches the LSF master
- NetApp Operations Manager 4.0 or later should be installed and monitoring the respective systems for LSF jobs
- Perl must be installed and running on the LSF master host

4 INSTALLATION

The installation and enablement of this solution are performed exclusively by NetApp Professional Services. Please talk to your NetApp sales representative for more details. The duration and pricing of the service may vary depending on the size and complexity of the environment and the level of integration required.

5 PLUG-IN COMPONENTS

The LSF Storage-Aware Plug-in includes the following components:

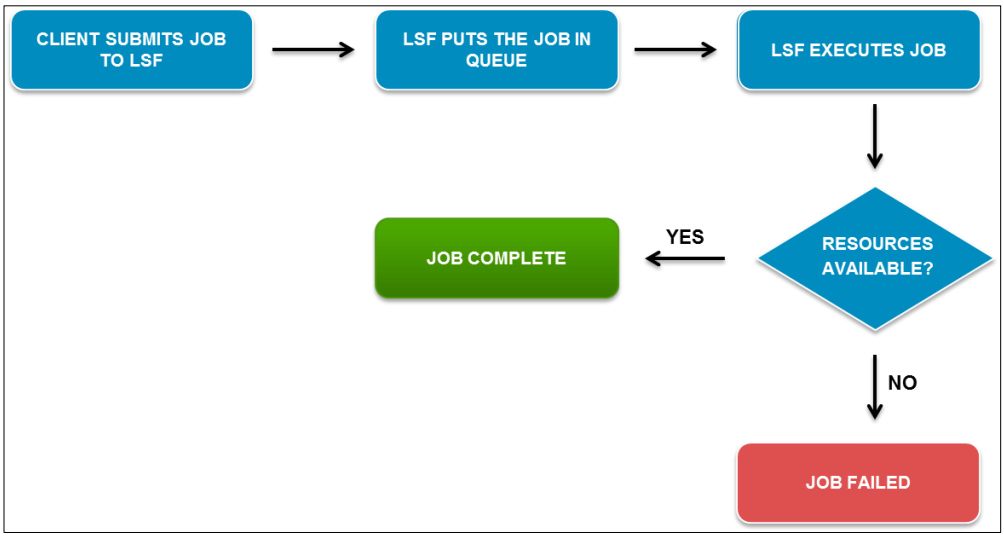
Table 1) Plug-in components.

Component	Purpose
schmod_filer.so	The scheduling plug-in implementing the job scheduling policy. Provided as a set of C source files that must be compiled.
filerplugin.conf	The plug-in configuration file. This file must be edited by the LSF administrator to configure mount points and appliances to be monitored.
elim.netapp	An LSF elim script that serves as a wrapper for the NetApp provided DataFabric® Manager script that gathers the load information from appliances.
getload	This is the NetApp provided script that retrieves appliance and volume load information from DataFabric Manager.
dfm_global.conf	Configuration file for the getload script.

6 SOLUTION WORKFLOW

When a job is submitted to LSF, the job waits in a queue until it is scheduled and executed. The time spent in the queue depends on the availability of requested resources. The LSF Storage-Aware Plug-in brings additional capabilities to monitor storage resources and execute jobs accordingly, instead of failing them.

Figure 2) LSF workflow without plug-in.



As you can see in **Figure 2**, LSF executes the job without checking for storage resources, but the success of the job completion actually depends on the availability of the storage resources. Even if the job starts successfully but the resources become unavailable in the middle of the job run, the job simply fails. However, with the plug-in, thresholds can be set for storage resources so that a job does not start until the threshold levels are satisfied, avoiding unsuccessful job runs and wasting storage resources.

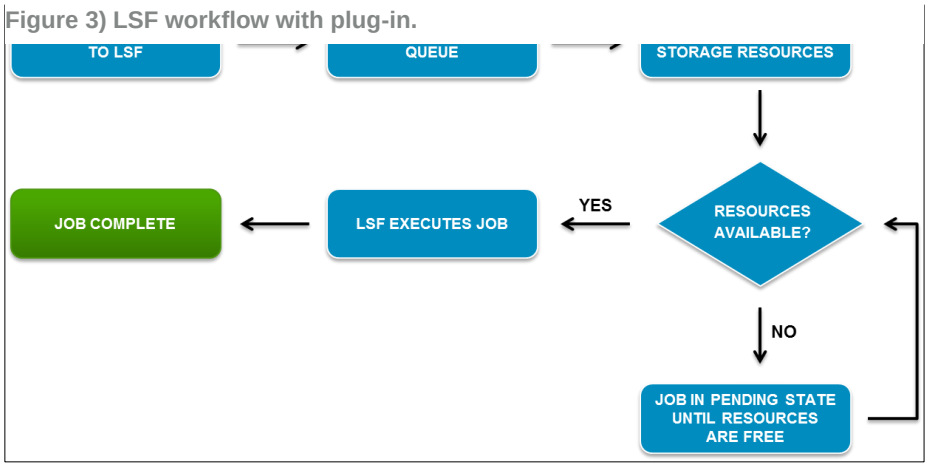
Figure 3 shows the workflow with the plug-in in place.

Best Practice
Set up threshold levels after carefully determining the requirements for the jobs. Because some jobs require more resources than others, threshold levels should be set so that most jobs are executed and few are placed in the PENDING state.

The LSF Storage-Aware Plug-in dispatches jobs to run based on the "load" on a set of NetApp appliances. LSF jobs indicate which mount points they use, and the plug-in determines whether:

- The file system at the mount point contains enough space, and
- Whether the appliance serving the mount point is overloaded or not and can keep jobs pending in the LSF queue.

Figure 3) LSF workflow with plug-in.



ONCE THE RESOURCES ARE AVAILABLE THE JOB IS EXECUTED AND SUCCESSFULLY COMPLETED.

7 HOW THE PLUG-IN WORKS

STEP 1: REPORTING OF STORAGE RESOURCES

Operations Manager is used to gather the storage metrics from the storage systems. Within Operations Manager a custom report is created that serves as the input for the LSF plug-in. The format of the report should conform to the guidelines set by the LSF Storage-Aware Plug-in. The following fields are required for the custom report.

Table 2) Operations Manager metrics.

Controller Metrics	Volume Metrics	Qtree Metrics
Controller ID	Volume ID	Qtree ID
Controller Full Name	Volume Full Name	Qtree Name
Controller Type	Volume Name	Qtree Fullname
Controller CPU %	Volume Used Capacity	Qtree Available %
Controller CPU Threshold	Volume Physical Capacity	Qtree Used Capacity %
Controller CPU Threshold Interval	Volume Used %	
	Volume Inode Used %	
	Volume Available %	

Any of these fields can be used as input for the LSF Storage-Aware Plug-in based on customer requirements. The threshold levels can be defined for any of these fields.

STEP 2: SETTING UP PLUG-IN PARAMETERS

Assuming LSF is installed and working, the LSF Storage-Aware Plug-in can be installed on the LSF master host. The following steps are involved in setting up the plug-in:

- Define the set of storage systems to monitor data collection.
- Determine what storage metrics need to be monitored.
- Set thresholds on each reported metric.
- Set up mount points for each LSF job.

Below is an example of how to set up the mount points in the **`filerplugin.conf`** file.

```
…
Begin ExportNames
FAS3270          /vol/LSF_VMCLONE        /LSF_VMCLONE
FAS3270          /vol/LSF_DATA           /LSF_DATA
FAS3270          /vol/vsc                /mnt
FAS6280          /vol/vmware_nfs         /vmware_nfs
FAS6280          /vol/vmware_nfs/q0      /vmware_nfs/q0
fas3170-lab14   /vol/vol0/vol0          /vol0
fas3170-lab14   /vol/fbsqlsysdb         /LSF_IMAGE
```

```
fas3170-lab14   /vol/oracle_arch1       /oracle_arch1
fas3170-lab14   /vol/fbsqltempdb        /fbsqltempdb
fas3170-lab14   /vol/cifs_test          /cifs_test
End ExportNames
…
```

Sample filerplugin.conf file

The first column shows the name of the NetApp system. The second column shows the export for the NetApp system. The third column indicates the mount point on the LSF system.

```
…
Begin PluginPolicy
FAS3270                            Max_Util            80
FAS3270:/vol/LSF_VMCLONE           Min_Avail_Space     40
FAS3270:/vol/LSF_DATA              Min_Avail_Space     50
FAS3270:/vol/vsc                   Min_Avail_Space     90
FAS6280                            Max_Util            90
FAS6280:/vol/vmware_nfs/q0         Min_Avail_Space     30
fas3170-lab14:/vol/vol0           Min_Avail_Space      30
fas3170-lab14                     Max_Util             80
fas3170-lab14:/vol/oracle_arch1   Min_Avail_Space      80
fas3170-lab14:/vol/fbsqltempdb    Min_Avail_Space      80
End PluginPolicy
…
```

Sample filerplugin.conf file

In line 1 of the example above, the system FAS3270 has a maximum CPU threshold of 80. This means that if the system has 80% CPU utilization, any job on that system will not run and instead will be placed in a PENDING state.

Line 2 of the above example shows that in addition to the CPU threshold, if the mount point **`/LSF_VMCLONE`** has less than 40% space available, the job will not run and instead will be placed in a PENDING state until at least 40% of the space is available.

STEP 3: JOB SUBMISSION

The LSF Storage-Aware Plug-in is activated for a job by using the bsub `-extsched` command-line option with the following syntax:

```
bsub -extsched "filer[/mount1 /mount2 ...]" ...
```

where /mount1, /mount2, etc., are the user-visible mount points that will be used by this job. At least one mount point must be listed, with multiple mount points separated by spaces.

A job will be kept pending if the amount of available space in the indicated mount point goes below the threshold defined in filerplugin.conf or the resource utilization of the appliance that exports the indicated mount point goes above the threshold defined in filerplugin.conf. If multiple mount points are specified, then the job will pend if any of the mount points cross the thresholds.

8 SCENARIOS SUCCESSFULLY TESTED

The following scenarios have been successfully tested with the LSF Storage-Aware Plug-in in place.

- Verify holding of a job in which a file system is below the available space threshold.
- Verify dispatch of a job in which a file system is above the available space threshold and the appliance's CPU utilization is under the defined threshold.
- Verify dispatch of a job in which 2 of 2 file systems are above the available space threshold.
- Verify holding of a job in which 1 of 2 file systems is below the available space threshold.
- Verify holding of a job in which 2 of 2 file systems are below the available space threshold.
- Verify holding of job in which 1 of 2 appliances has CPU utilization above the threshold. Each listed file system comes from a different appliance, and both are above the available space threshold (that is, based on space, they would be dispatched).
- Submit a 10-element job array (that is, 10 jobs) with the same requirement. As they run, cause the file system to go below the available space threshold and show that subsequent jobs are being held. Once the resource is below the threshold level, show the jobs being dispatched again.
- Show how to disable the plug-in without unconfiguring it from LSF. Submit a job that the plug-in causes to hold and then disable the plug-in, causing the job to be dispatched. Also show reenabling.

In addition, the LSF Storage-Aware Plug-in can be customized to fit customer needs. Please talk to a NetApp sales representative if you have a question about a specific scenario.

9 NEXT STEPS

There has been a huge amount of interest in this solution from various customers. At the time this document was created, this solution was being deployed at a major ISV along with multiple demonstrations at various customer sites. Please contact the author of this paper (saif.hameed@netapp.com) to find out the status of those deployments or more information regarding this solution.

10 FREQUENTLY ASKED QUESTIONS

Q. Where do I look for errors?

A. Errors or inconsistencies in the filerplugin.conf file will be logged in filerplugin.log in the Work_Dir. With debugging turned on, filerplugin.log will contain messages about scheduling decisions (that is, whether a job has been kept pending or was dispatched).

Q. Is it possible to temporarily disable the plug-in?

A. A simple way to disable the plug-in without reversing all of the configuration described above is to set an incorrect password for the DataFabric Manager user in dfm_global.conf and to remove the "load" file in Work_Dir. Reenable the plug-in by setting the password correctly again.